

Języki Programowania, środa 14.10.2020

Zadanie 1

(1) Należy otworzyć pusty plik tekstowy - powinien mieć rozszerzenie `.cpp`. Należy dołączyć biblioteki wejścia/wyjścia:

```
#include <iostream>
```

zdefiniować przestrzeń nazw:

```
using namespace std;
```

oraz napisać funkcję `int main()`. W funkcji wypisać na ekran napis „Wektor (5,6)”.

Skompilować i uruchomić.

Pamiętajcie, żeby regularnie zapisywać i kompilować Wasze programy!

Wczytać dwie liczby z klawiatury. Wypisać je na ekran podobnie jak powyższe „5” i „6”.

(2) Należy stworzyć strukturę **Wektor** reprezentującą wektor w przestrzeni dwuwymiarowej. Struktura ta powinna zawierać dwa pola typu `double` (składowe `x` i `y`):

- `double x;`

- `double y;`

oraz zawierać następujące metody (funkcje składowe klasy):

- `void zapisz(double xx, double yy)` - umożliwiającą zapisanie danego wektora

- `void wypisz()` - wypisującą dany wektor na ekran w formacie `[x, y]`

W programie głównym należy stworzyć pojedynczy obiekt typu `Wektor` (`x = 1, y = 4`) - używając metody `zapisz`, i wypisać go na ekran - używając metody `wypisz`.

Następnie wczytać dwie liczby z klawiatury, odpowiadające `x` oraz `y` i stworzyć drugi wektor z podanymi wartościami, również wypisać go na ekran (używając obu metod).

(3) Proszę przerobić strukturę na klasę. Słowo kluczowe `struct` zamienić na słowo kluczowe `class`

Wszystkie składniki i metody mogą być publiczne (`public`).

- **Proszę sprawdzić, jaki efekt ma NIE umieszczenie słowa kluczowego „public” w kodzie klasy. Proszę efekt zapisać w komentarzu w kodzie.**

(4) Dopisać funkcję składową **zwracającą** długość wektora na którym wywołana jest metoda:

[**typ zwracany**] `dlugosc()` | by użyć pierwiastka kwadratowego (`sqrt`) należy załączyć bibliotekę `math.h`:

```
#include <math.h>
```

Przetestować ją w funkcji `main()` wypisując na ekran długość wektora (5,6).

UWAGA! Jeśli w treści zadania pojawia się prośba o napisanie funkcji która „zwraca” wartość, funkcja powinna tę wartość zwracać (poprzez `return` bądź parametr, a nie wypisywać na ekran!)

(5) Klasa powinna zostać podzielona na dwa oddzielne pliki: `wektor.h` i `wektor.cpp`.

W pliku `*.h` powinna się znaleźć definicja klasy (słowo kluczowe `class`, składniki klasy, oraz deklaracje funkcji/metod), natomiast w pliku `*.cpp` – ciało funkcji składowych (metod). Dodatkowo należy stworzyć główny plik programu zawierający funkcję `main()` - **razem 3 pliki**.

(6) Dopisać następującą funkcję składową:

- *dodając wielkości liczbowe = zwiększające wartości składowych wektora:*

```
void dodaj(double xx, double yy)
```

- *przypisując wartości składowych wektora z innego wektora (zmieniający wartości `x` i `y` wektora na którym funkcja jest wywoływana z innego wektora podanego jako argument funkcji):*

```
void kopiuj(Wektor w)
```

Przetestować obie w funkcji `main()` wypisując na ekran wektor pierwszy po dodaniu do niego parametrów (9,9) i wektor drugi, po przypisania mu wartości z wektora pierwszego.

Wskazówki:

- By dostać się do składników struktury „na zewnątrz” używamy ".", np. `cout<<A.F()<<endl;` (jeśli są publiczne)
- Należy pamiętać o załączeniu biblioteki `punkt.h` w plikach `*.cpp` (`#include "wektor.h"`).
- W pliku `punkt.cpp` przy definicjach funkcji należy pamiętać o dodaniu operatora zakresu `"wektor : :"`